**Exercise 3: Sorting Customer Orders**

**Scenario:**

**You are tasked with sorting customer orders by their total price on an e-commerce platform. This helps in prioritizing high-value orders.**

**Steps:**

1. **Understand Sorting Algorithms:**

* **Explain different sorting algorithms (Bubble Sort, Insertion Sort, Quick Sort, Merge Sort).**

1. Bubble Sort: This algorithm repeatedly iterates through the list, comparing each pair of adjacent elements and swapping them if they are in the wrong order. This process continues until the list is sorted.
2. Insertion Sort: A straightforward sorting method that builds the final sorted list one item at a time. It inserts each element from an unsorted portion into its correct position within a sorted portion. It is a stable, in-place algorithm that maintains the relative order of equal elements and does not require additional memory beyond the original array.
3. Quick Sort: This sorting algorithm uses the Divide and Conquer approach. It selects a pivot element, partitions the array around the pivot, and places the pivot in its correct position in the sorted array.
4. Merge Sort: A sorting method that follows a divide-and-conquer strategy. It recursively divides the array into smaller subarrays, sorts these subarrays, and then merges them back together to form a sorted array.
5. **Setup:**
   * **Create a class Order with attributes like orderId, customerName, and totalPrice.**
6. **Implementation:**
   * **Implement Bubble Sort to sort orders by totalPrice.**
   * **Implement Quick Sort to sort orders by totalPrice.**
7. **Analysis:**

* **Compare the performance (time complexity) of Bubble Sort and Quick Sort.**

1. Bubble Sort :-

* Best Case: O(n)
* Average Case: O(n^2)
* Worst Case: O(n^2)

1. Quick Sort :-

* Best Case: O(n log n)
* Average Case: O(n log n)
* Worst Case: O(n^2)
* **Discuss why Quick Sort is generally preferred over Bubble Sort.**

1. Quick Sort has an average time complexity of O(n logn) which is much more efficient than Bubble Sort’s average time complexity of O(n^2).
2. As a divide-and-conquer algorithm, Quick Sort performs better on larger datasets compared to Bubble Sort.